[55] W. Fu, R. Lin, and D. Inge, "TaintAssembly: Taint-based Information Flow Control Tracking for WebAssembly," *CoRR*, vol. abs/1802.01050, 2018.

[56] Q. Stiévenart, D. Binkley, and C. De Roover, "Dynamic Slicing of WebAssembly Binaries," in *39th IEEE International Conference on Software Maintenance and Evolution*, IEEE, 2023.

[57] Q. Stiévenart, D. W. Binkley, and C. D. Roover, "Static Stack-preserving Intra-procedural Slicing of WebAssembly Binaries," in *44th IEEE/ACM 44th International Conference on Software Engineering, ICSE 2022, Pittsburgh, PA, USA, May 25-27, 2022*, pp. 2031–2042, ACM, 2022.

[58] D. Lehmann and M. Pradel, "Wasabi: A Framework for Dynamically Analyzing WebAssembly," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS*, pp. 1045–1058, 2019.

[59] S. Narayan, C. Disselkoen, D. Moghimi, S. Cauligi, E. Johnson, Z. Gang, A. Vahldiek-Oberwagner, R. Sahita, H. Shacham, D. M. Tullsen, and D. Stefan, "Swivel: Hardening WebAssembly against Spectre," in *30th USENIX Security Symposium, USENIX*, pp. 1433–1450, 2021.

[60] E. Johnson, E. Laufer, Z. Zhao, D. Gohman, S. Narayan, S. Savage, D. Stefan, and F. Brown, "WaVe: A Verifiably Secure WebAssembly Sandboxing Runtime," in *44th IEEE Symposium on Security and Privacy, SP 2023, San Francisco, CA, USA, May 21-25, 2023*, pp. 2940–2955, IEEE, 2023.

[61] M. Musch, C. Wressnegger, M. Johns, and K. Rieck, "New Kid on the Web: A Study on the Prevalence of WebAssembly in the Wild," in *Detection of Intrusions and Malware, and Vulnerability Assessment - 16th International Conference, DIMVA*, vol. 11543, pp. 23–42, 2019.

[62] S. Bhansali, A. Aris, A. Acar, H. Oz, and A. S. Uluagac, "A First Look at Code Obfuscation for WebAssembly," in *WiSec '22: 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 140–145, 2022.

[63] B. Baudry and M. Monperrus, "The Multiple Facets of Software Diversity: Recent Developments in Year 2000 and Beyond," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 16:1–16:26, 2015.

[64] K. Pohl, G. Böckle, and F. van der Linden, *Software Product Line Engineering - Foundations, Principles, and Techniques.* Springer, 2005.

[65] S. Sidiroglou-Douskos, S. Misailovic, H. Hoffmann, and M. C. Rinard, "Managing Performance vs. Accuracy Trade-offs With Loop Perforation," in *SIGSOFT/FSE'11 19th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-19) and ESEC'11: 13th European Software Engineering Conference (ESEC-13)*, pp. 124–134, 2011.

[66] Avizienis and Kelly, "Fault Tolerance by Design Diversity: Concepts and Experiments," *Computer*, vol. 17, no. 8, pp. 67–80, 1984.

[67] T. Y. Chen, F. Kuo, R. G. Merkel, and T. H. Tse, "Adaptive Random Testing: The ART of test case diversity," *J. Syst. Softw.*, vol. 83, no. 1, pp. 60–66, 2010.

[68] T. Jackson, *On the Design, Implications, and Effects of Implementing Software Diversity for Security.* PhD thesis, University of California, Irvine, 2012.

[69] G. R. Lundquist, V. Mohan, and K. W. Hamlen, "Searching for Software0 Diversity: Attaining Artificial Diversity through Program Synthesis," in *Proceedings of the 2016 New Security Paradigms Workshop*, NSPW '16, (New York, NY, USA), p. 80–91, Association for Computing Machinery, 2016.

[70] B. Randell, "System Structure for Software Fault Tolerance," *SIGPLAN Not.*, vol. 10, p. 437–449, apr 1975.

[71] N. Harrand, *Software Diversity for Third-Party Dependencies.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2022.

[72] J. V. Cleemput, B. Coppens, and B. D. Sutter, "Compiler Mitigations for Time Attacks on Modern X86 Processors," *ACM Trans. Archit. Code Optim.*, vol. 8, no. 4, pp. 23:1–23:20, 2012.

[73] A. Homescu, S. Neisius, P. Larsen, S. Brunthaler, and M. Franz, "Profile-guided Automated Software Diversity," in *Proceedings of the 2013 IEEE/ACM International Symposium on Code Generation and Optimization, CGO 2013, Shenzhen, China, February 23-27, 2013*, pp. 23:1–23:11, IEEE Computer Society, 2013.

[74] S. Bhatkar, D. C. DuVarney, and R. Sekar, "Address Obfuscation: An Efficient Approach to Combat a Board Range of Memory Error Exploits," in *Proceedings of the USENIX Security Symposium*, 2003.

[75] S. Bhatkar and D. C. DuVarney, "Efficient Techniques for Comprehensive Protection from Memory Error Exploits," in *Proceedings of the 14th USENIX*, 2005.

[76] K. Pettis and R. C. Hansen, "Profile Guided Code Positioning," in *Proceedings of the ACM SIGPLAN'90 Conference on Programming Language Design and Implementation (PLDI)*, pp. 16–27, 1990.

[77] S. Crane, A. Homescu, S. Brunthaler, P. Larsen, and M. Franz, "Thwarting Cache Side-channel Attacks Through Dynamic Software Diversity," in *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015*, The Internet Society, 2015.